# Tips for Bcfg2 Development

1. Focus on either the client or server code. This focuses the development process down the the precise pieces of code that matter for the task at hand.

- If you are developing a client driver, then write up a small configuration specification that includes the needed characteristics.
- If you are working on the server, run bcfg2−info and use to assess the code

2. Use the python interpreter. One of python's most appealing features is interactive use of the interpreter.

- If you are developing for the client−side, run "python −i /usr/sbin/bcfg2" with the appropriate bcfg2 options. This will cause the python interpreter to continue running, leaving all variables intact. This can be used to examine data state in a convenient fashion.
- If you are developing for the server side, use bcfg2−info and the "debug" option. This will leave you at a python interpreter prompt, with the server core loaded in the variable "bcore".

3. Use pylint obsessively. It raises a lot of style−related warnings which can be ignored, but most all of the errors are legitimate.

4. If you are doing anything with Regular Expressions, Kodos − The Python Regular Expression Debugger is your friend.